Info IV Tutorium

Mit Vertretung für Tut 7 :-)

Thomas Pajor

Fakultät für Informatik

IBDS Prautzsch

29. Mai 2007

Organisatorisches

Bedingt durch den Urlaub von Bertrand Klimmek gibt es ein paar Änderungen im Übungsbetrieb.

- Das Übungsblatt Nr. 5 gibt es erst nächste Woche zurück

Organisatorisches

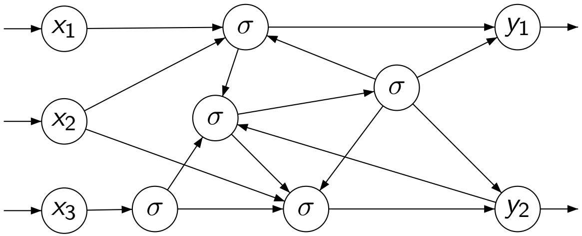
Bedingt durch den Urlaub von Bertrand Klimmek gibt es ein paar Änderungen im Übungsbetrieb.

- Das Übungsblatt Nr. 5 gibt es erst nächste Woche zurück
- Für das 6. Übungsblatt habt ihr zwei Wochen Zeit. Abgabe ist erst am 6. Juni

Definition

Ein *Neuronales Netz* (NN) besteht aus einer Menge von *Neuronen* (Knoten) und einer Menge von gerichteten *Kanten* (Synapsen). Jeder Kante zwischen den Neuronen N_i und N_j ist ein *Gewicht* w_{ij} zugeordnet. Jedes Neuron N_j hat außerdem einen *Schwellwert* σ_j.

Neuronale Netze werden meistens zur *Klassifikation* eingesetzt.

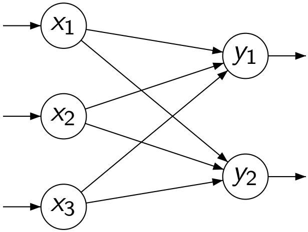


Wiederholung: Neuronale Netze

Perzeptron

Das *Perzeptron* ist ein vorwärtsgerichtetes Neuronales Netz, bei dem die Neuronen schichtweise angeordnet sind. Verbindungen existieren nur zwischen benachbarten Schichten.

Das *einfache Perzeptron* besitzt nur eine Eingabe- sowie eine Ausgabeschicht.



Arbeitsweise der Neuronen

Die Neuronen arbeiten alle *synchron* in diskreten Zeitabständen. Jedes Neuron N_j hat einen *Zustand* (bzw. eine Ausgabe) $q_j \in Q$, der an den Ausgabekanten anliegt.

Die *Erregungsfunktion* p_j des Neurons N_j berechnet sich aus den Werten der Eingangskanten sowie dem Schwellwert von N_j:

$$p_j := \sum_i w_{ij} q_i - \sigma_j$$

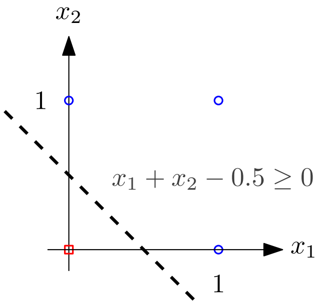
Der *Zustand* q_j des Neurons N_j im nächsten Zeittakt ist dann für $Q := \{0, 1\}$

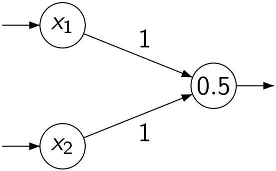
$$q_j := \begin{cases} 1 & \text{falls} \quad p_j \geq 0 \\ 0 & \text{sonst} \end{cases}$$

Wiederholung: Neuronale Netze

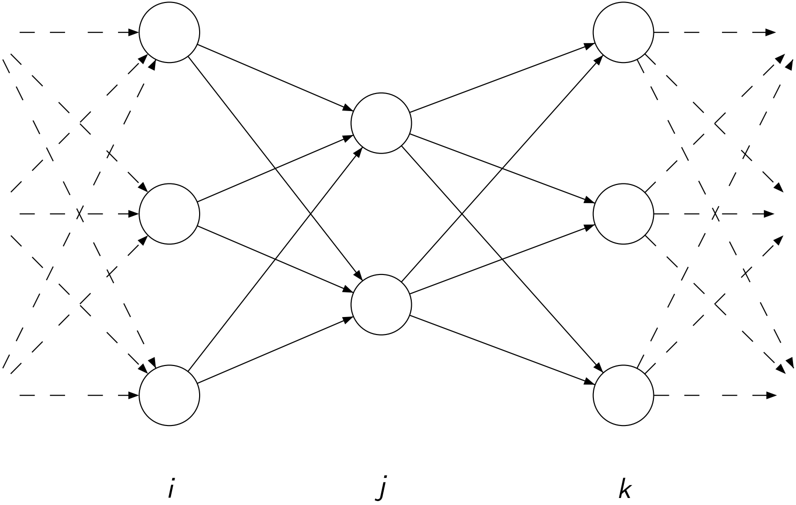
Mächtigkeit des einfachen Perzeptrons

Mit dem einfachen Perzeptron können genau die Klassifikationsprobleme gelöst werden, die *linear separierbar* sind, i.e. bei denen eine Hyperebene die Klassen voneinander trennt.





Bezeichnung der Variablen



Backpropagation

Backpropagation

- Erregungsfunktion für ein Neuron N_j

$$p_j = \sum_i w_{ij} q_i$$

Backpropagation

- Erregungsfunktion für ein Neuron N_j und zu minimierender Fehler:

$$p_j = \sum_i w_{ij} q_i \qquad\qquad E(W) = \frac{1}{2} \sum_j (a_j - q_j)^2$$

Backpropagation

- Erregungsfunktion für ein Neuron N_j und zu minimierender Fehler:

$$p_j = \sum_i w_{ij} q_i \qquad\qquad E(W) = \frac{1}{2} \sum_j (a_j - q_j)^2$$

- Gradientenabstieg:

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

Backpropagation

- Erregungsfunktion für ein Neuron N_j und zu minimierender Fehler:

$$p_j = \sum_i w_{ij} q_i \qquad E(W) = \frac{1}{2} \sum_j (a_j - q_j)^2$$

- Gradientenabstieg:

$$w_{ij} = w_{ij} + \Delta w_{ij} = w_{ij} - \eta \cdot \frac{\partial E}{\partial w_{ij}}$$

Backpropagation

- Erregungsfunktion für ein Neuron N_j und zu minimierender Fehler:

$$p_j = \sum_i w_{ij} q_i \qquad\qquad E(W) = \frac{1}{2} \sum_j (a_j - q_j)^2$$

- Gradientenabstieg:

$$w_{ij} = w_{ij} + \Delta w_{ij} = w_{ij} - \eta \cdot \frac{\partial E}{\partial w_{ij}}$$

- Fehler:

$$\delta_j := -\frac{\partial E}{\partial p_j}$$

Backpropagation

- Erregungsfunktion für ein Neuron N_j und zu minimierender Fehler:

$$p_j = \sum_i w_{ij} q_i \qquad\qquad E(W) = \frac{1}{2} \sum_j (a_j - q_j)^2$$

- Gradientenabstieg:

$$w_{ij} = w_{ij} + \Delta w_{ij} = w_{ij} - \eta \cdot \frac{\partial E}{\partial w_{ij}} = w_{ij} + \underbrace{\eta \delta_j q_i}_{=\Delta w_{ij}}$$

- Fehler:

$$\delta_j := -\frac{\partial E}{\partial p_j}$$

Backpropagation

- Erregungsfunktion für ein Neuron N_j und zu minimierender Fehler:

$$p_j = \sum_i w_{ij} q_i \qquad E(W) = \frac{1}{2} \sum_j (a_j - q_j)^2$$

- Gradientenabstieg:

$$w_{ij} = w_{ij} + \Delta w_{ij} = w_{ij} - \eta \cdot \frac{\partial E}{\partial w_{ij}} = w_{ij} + \underbrace{\eta \delta_j q_i}_{=\Delta w_{ij}}$$

- Fehler:

$$\delta_j := -\frac{\partial E}{\partial p_j} = -\frac{\partial E}{\partial q_j} \frac{\partial q_j}{\partial p_j}$$

Backpropagation

- Erregungsfunktion für ein Neuron N_j und zu minimierender Fehler:

$$p_j = \sum_i w_{ij} q_i \qquad\qquad E(W) = \frac{1}{2} \sum_j (a_j - q_j)^2$$

- Gradientenabstieg:

$$w_{ij} = w_{ij} + \Delta w_{ij} = w_{ij} - \eta \cdot \frac{\partial E}{\partial w_{ij}} = w_{ij} + \underbrace{\eta \delta_j q_i}_{=\Delta w_{ij}}$$

- Fehler:

$$\delta_j := -\frac{\partial E}{\partial p_j} = -\frac{\partial E}{\partial q_j} \frac{\partial q_j}{\partial p_j}$$

- Berechnung von $-\partial E / \partial q_j$:

Ausg.: $-\dfrac{\partial E}{\partial q_j} = (a_j - q_j)$

Backpropagation

- Erregungsfunktion für ein Neuron N_j und zu minimierender Fehler:

$$p_j = \sum_i w_{ij} q_i \qquad E(W) = \frac{1}{2} \sum_j (a_j - q_j)^2$$

- Gradientenabstieg:

$$w_{ij} = w_{ij} + \Delta w_{ij} = w_{ij} - \eta \cdot \frac{\partial E}{\partial w_{ij}} = w_{ij} + \underbrace{\eta \delta_j q_i}_{=\Delta w_{ij}}$$

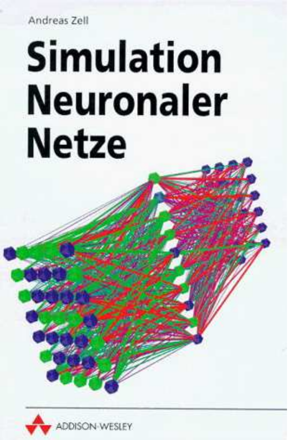
- Fehler:

$$\delta_j := -\frac{\partial E}{\partial p_j} = -\frac{\partial E}{\partial q_j} \frac{\partial q_j}{\partial p_j}$$

- Berechnung von $-\partial E / \partial q_j$:

$$\text{Ausg.:} \ -\frac{\partial E}{\partial q_j} = (a_j - q_j) \qquad \text{Inn. Schicht:} \ -\frac{\partial E}{\partial q_j} = \sum_k \delta_k w_{jk}$$

Buchempfehlung von mir



Andreas Zell - *Simulation Neuronaler Netze*
2. Auflage Oldenbourg Verlag
ISBN: 3-486-24350-0

1. Auflage
Addison Wesley Verlag

Auch in der Uni-Bib erhältlich!